Prediction of Medical Expenses of Individuals using Regression Models

Parvez Khan

Table of Contents

Dataset....1
Objective....1
Data description....1
Data Cleaning....2
Data Visualization....2
Model Fitting....14
Conclusion....19

Dataset

Here we have a dataset about medical costs billed by health insurance on different individuals along with their age, sex, bmi, number of children and other parameters collected from kaggle database.

Objective

To predict the medical bill for individuals based on different parameters using different regression models and choose the best among them.

Data description

Loading the data

```
data <- read.csv("insurance.csv", header = TRUE)
```

Checking for number of rows and columns

```
dim(data)
```

```
## [1] 1338    7
```

Taking a look at the data frame

```
head(data)
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
## 6  31 female 25.740        0     no southeast  3756.622
```

```
str(data)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : chr  "female" "male" "male" "male" ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : chr  "yes" "no" "no" "no" ...
##  $ region  : chr  "southwest" "southeast" "southeast" "northwest" ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

Data Cleaning

Checking for missing values

```
missing <- sum(is.na(data))
missing
```

```
## [1] 0
```

Comment: There is no missing value in the data.

Checking for duplicate values

```
duplicate_rows <- data[duplicated(data),]
duplicate_rows
```

```
##     age  sex   bmi children smoker    region  charges
## 582  19 male 30.59        0     no northwest 1639.563
```

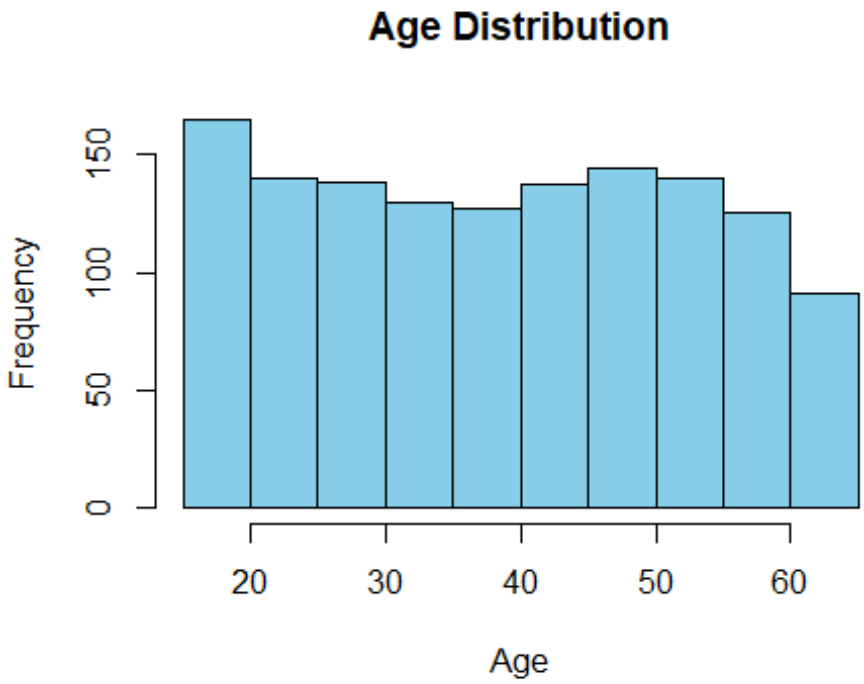
Comment: There is one duplicate row, so we remove it from the data.

```
data <- data[!duplicated(data),]
```

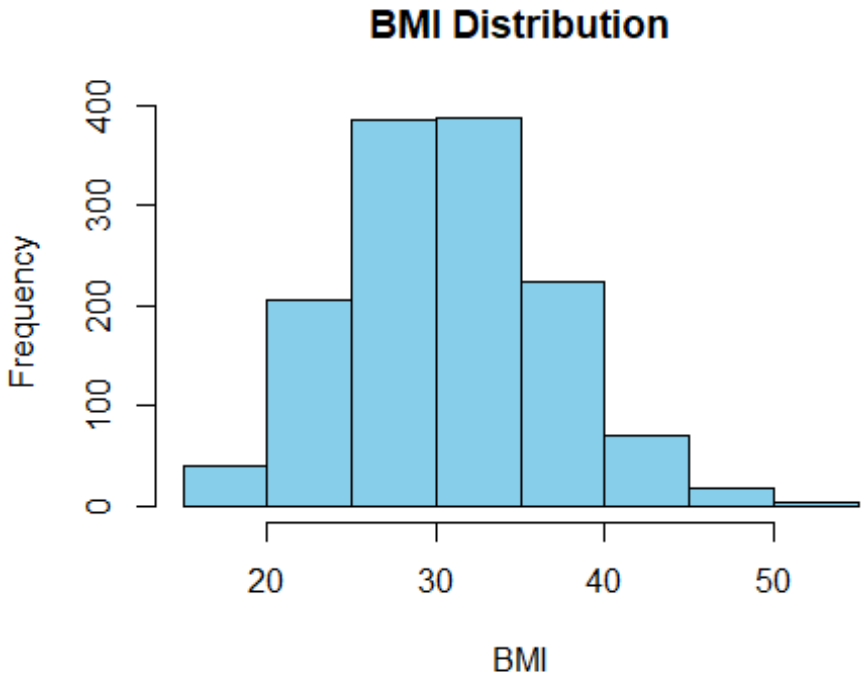
Data Visualization

To start with data visualization we first plot histograms for the numeric columns

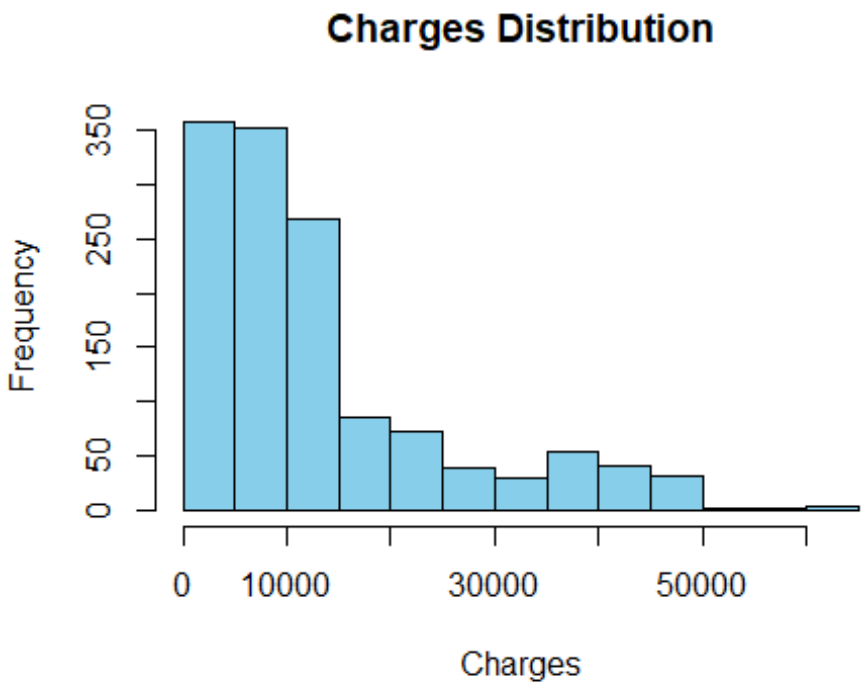
```
hist(data$age, main = "Age Distribution", xlab = "Age", ylab = "Frequency",
col = "skyblue")
```




```r
hist(data$bmi, main = "BMI Distribution", xlab = "BMI", ylab = "Frequency",
col = "skyblue")
```



```
hist(data$charges, main = "Charges Distribution", xlab = "Charges", ylab =
"Frequency", col = "skyblue")
```



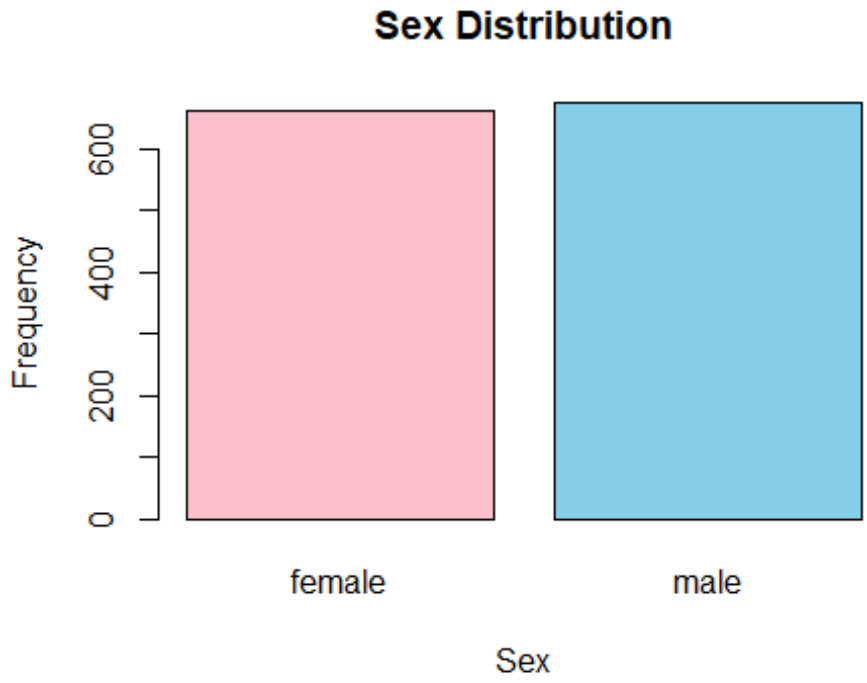
Now for the categorical columns first we will get the number of occurrence for each unique entries

```
sex_count <- table(data$sex)
sex_count

##
## female   male
##    662    675

children_count <- table(data$children)
children_count

##
##   0   1   2   3   4   5
## 573 324 240 157  25  18

smoker_count <- table(data$smoker)
smoker_count

##
##   no  yes
## 1063  274
```

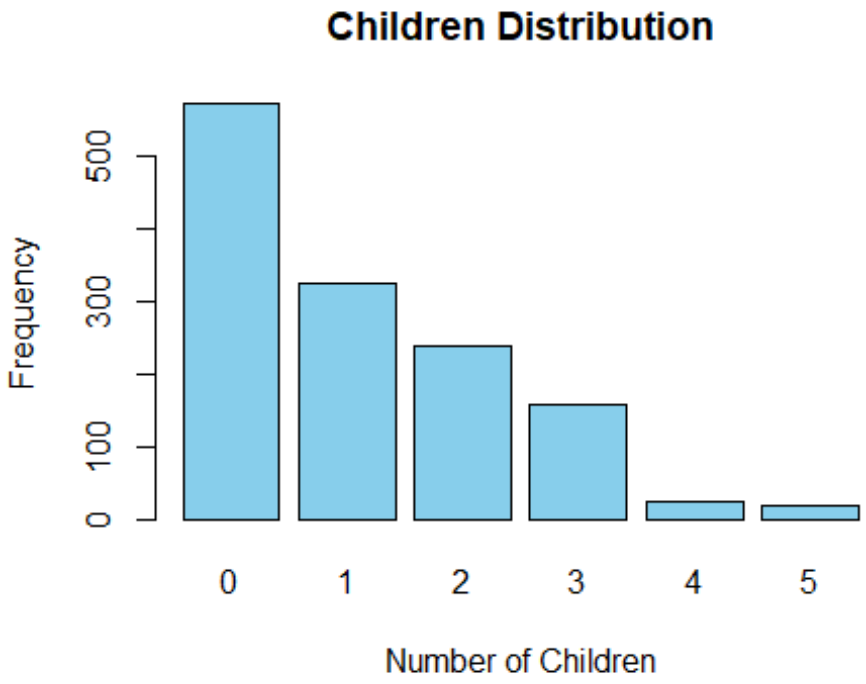
```
region_count <- table(data$region)
region_count

##
## northeast northwest southeast southwest
##       324       324       364       325
```

Now to visualize the distribution of categorical columns we will plot their bar graph

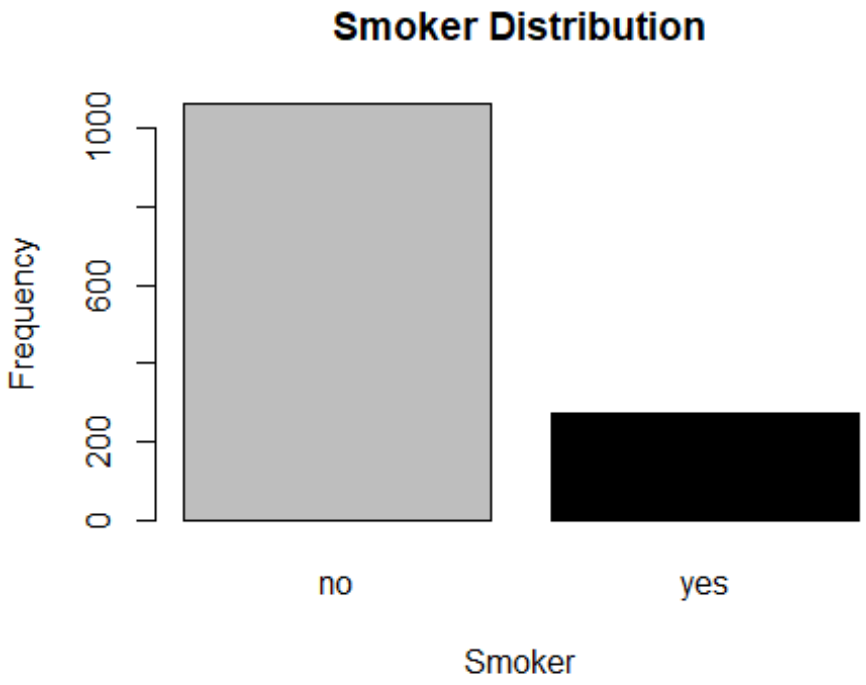
```
barplot(sex_count, main = "Sex Distribution", xlab = "Sex", ylab =
"Frequency", col = c("pink", "skyblue"))
```



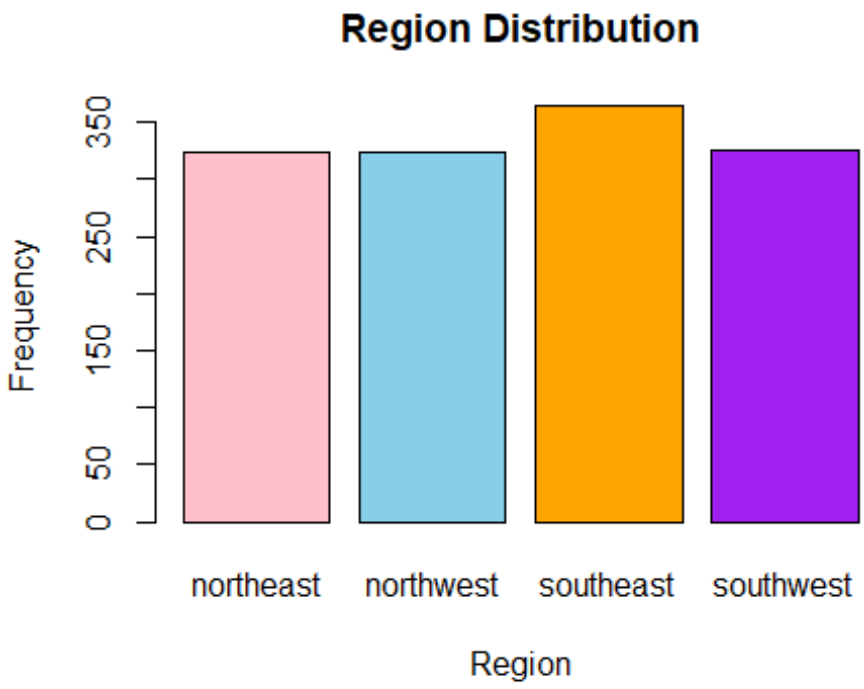

```
barplot(children_count, main = "Children Distribution", xlab = "Number of
Children", ylab = "Frequency", col = "skyblue")
```




```r
barplot(smoker_count, main = "Smoker Distribution", xlab = "Smoker", ylab =
"Frequency", col = c("grey", "black"))
```

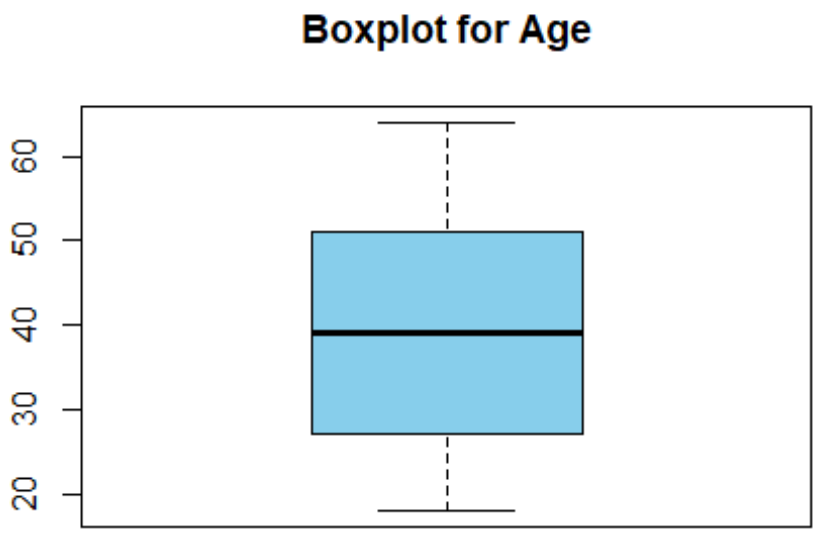


```r
barplot(region_count, main = "Region Distribution", xlab = "Region", ylab =
"Frequency", col = c("pink", "skyblue", "orange", "purple"))
```

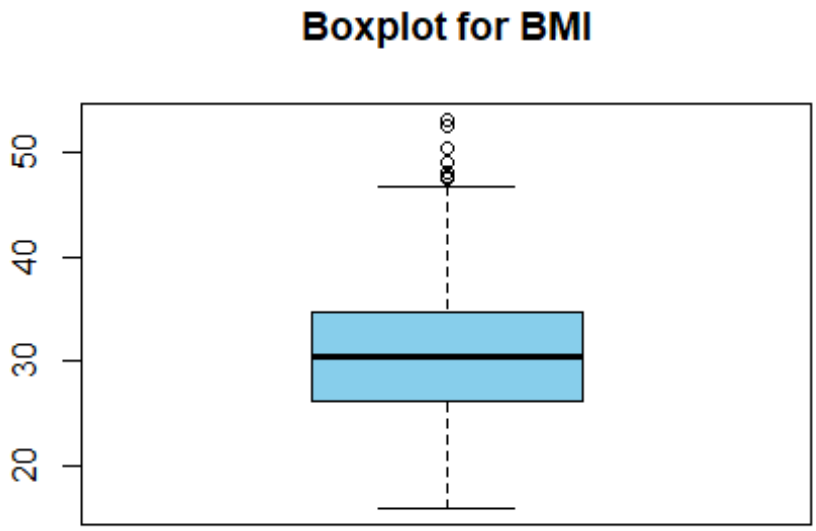


Now we plot the boxplot for numerical columns to check for outliers

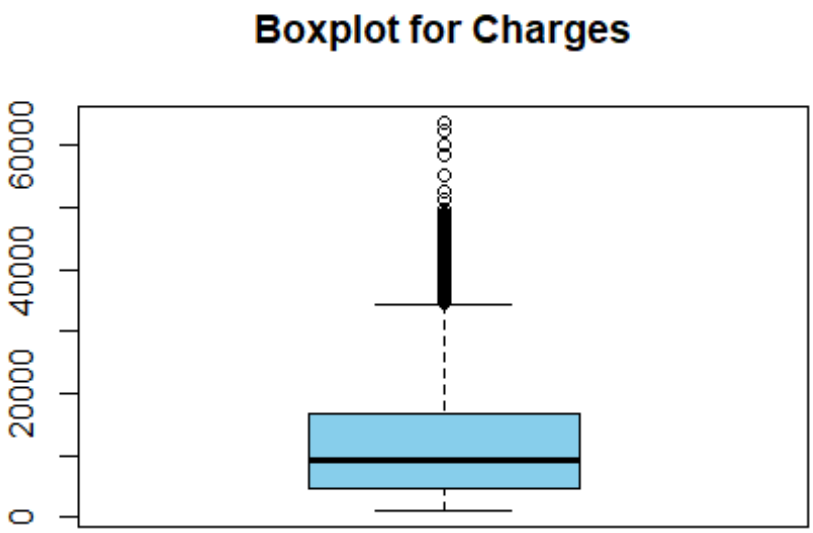
```r
boxplot(data$age, main = "Boxplot for Age", col = "skyblue")
```




```r
boxplot(data$bmi, main = "Boxplot for BMI", col = "skyblue")
```



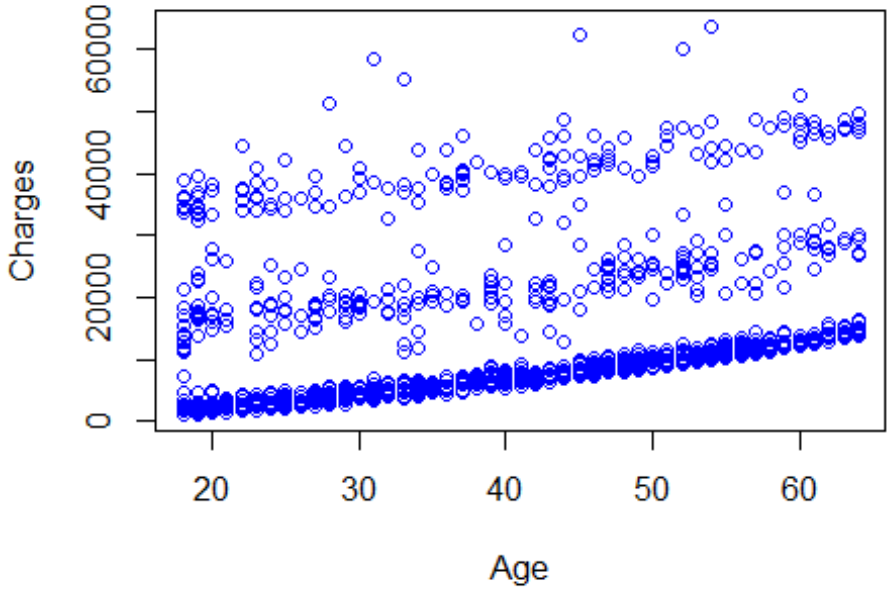

```r
boxplot(data$charges, main = "Boxplot for Charges", col = "skyblue")
```



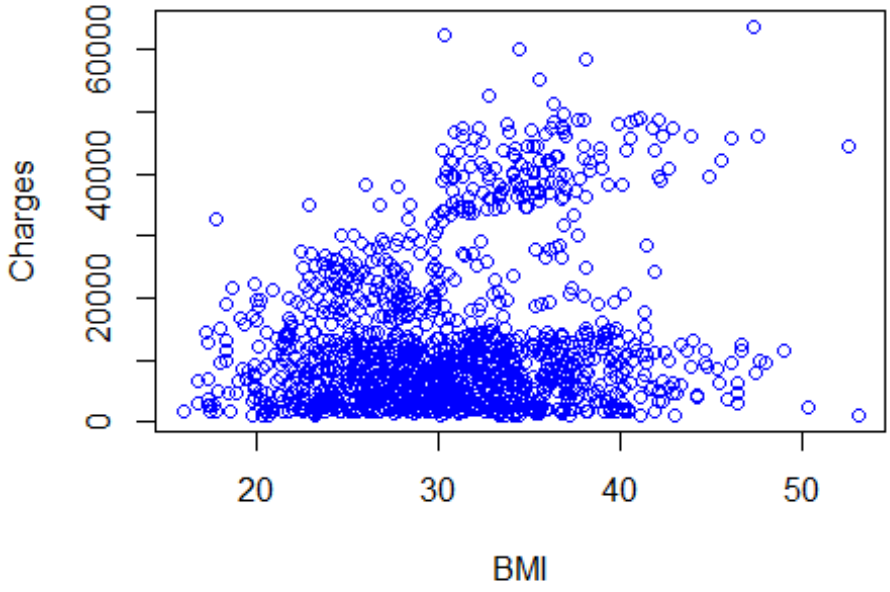
Comment: It is evident from the boxplot that bmi and chrages contains some outliers while there is no outliers on the age column.

Now we plot a scatter diagram for further interpretation

```
plot(data$age, data$charges, xlab = "Age", ylab = "Charges", col = "blue")
```




```r
plot(data$bmi, data$charges, xlab = "BMI", ylab = "Charges", col = "blue")
```



Comment: From the scatter plot it can be seen that their is a weak relation between age and charges while the relation is more weak between bmi and charges.

Now we create a correlation matrix to further check for quantified relationship between the variables.

```
numeric_data <- data[, c("age", "bmi", "charges")]
cor(numeric_data)
```

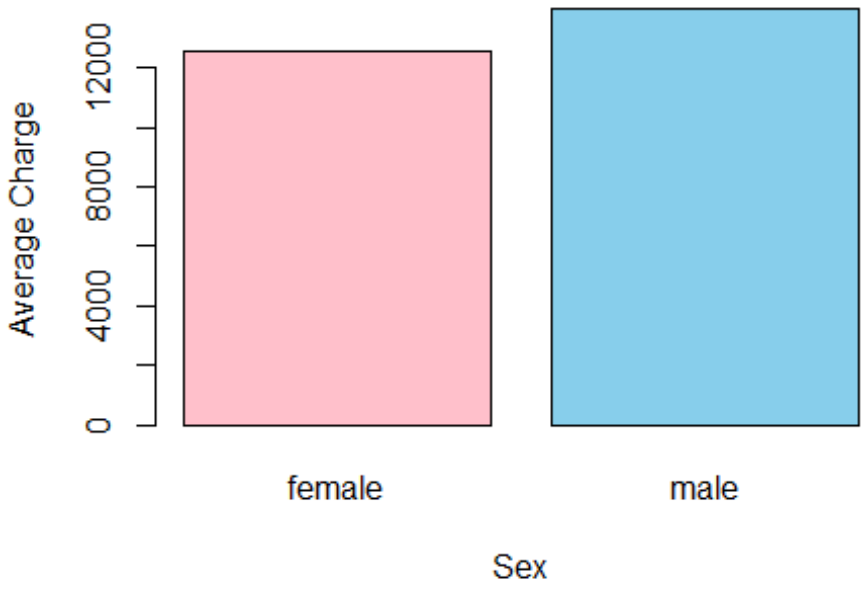
```
##                age       bmi   charges
## age     1.0000000 0.1093436 0.2983082
## bmi     0.1093436 1.0000000 0.1984008
## charges 0.2983082 0.1984008 1.0000000
```

Comment: From the correlation matrix it is evident that the correlation between bmi and charges is very low while between age and charges it is slightly better.

Now we see if the charges vary with different categories

Charges for males and females:

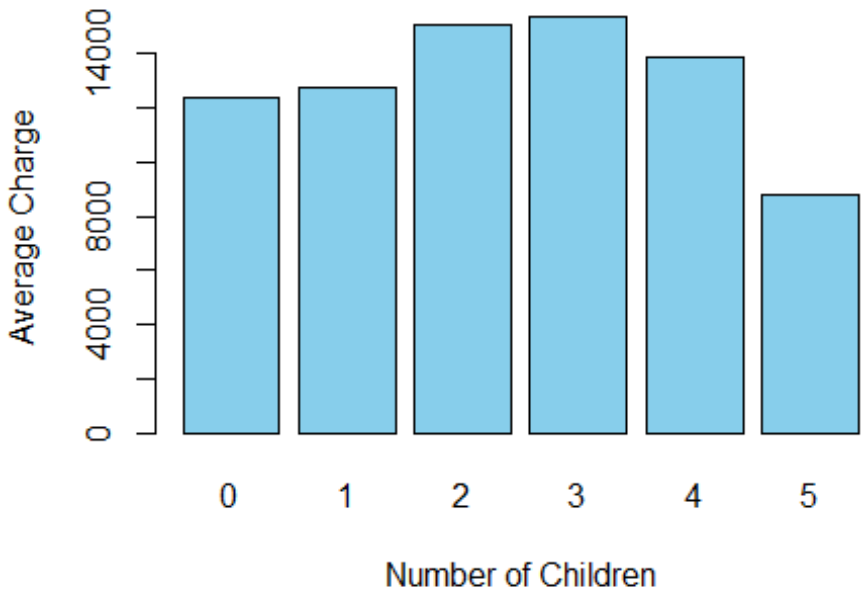
```
av_sex_charge <- tapply(data$charges, data$sex, mean)
barplot(av_sex_charge, xlab = "Sex", ylab = "Average Charge", col = c("pink",
"skyblue"))
```



Comment: As it can be seen there is not much difference in medical charge of male and female.

Charges for different number children:

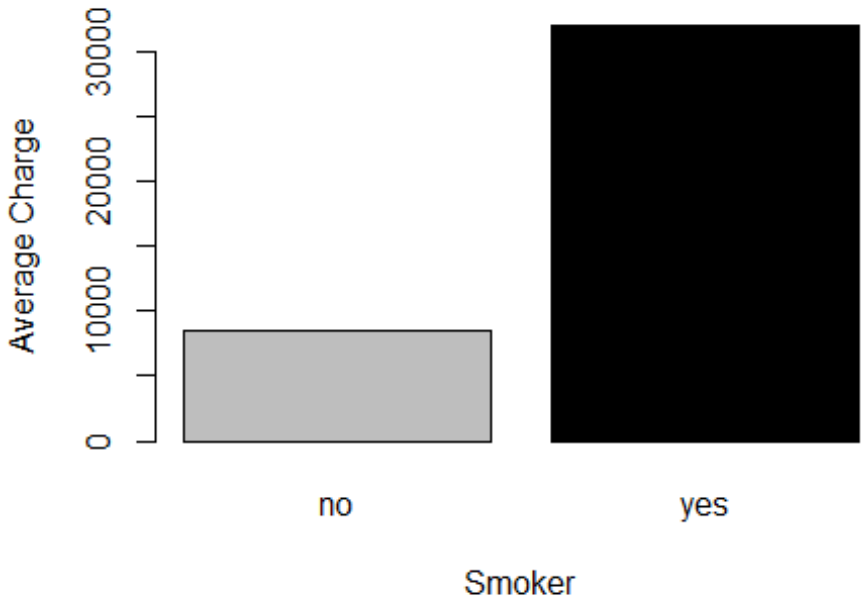
```
av_children_charge <- tapply(data$charges, data$children, mean)
barplot(av_children_charge, xlab = "Number of Children", ylab = "Average
Charge", col = "skyblue")
```



Comment: From the plot we can see that the average charge is almost not very different among people with different number of children with an exception where surprisingly people with 5 children has less average medical charge.

Charges for smokers and non-smokers:

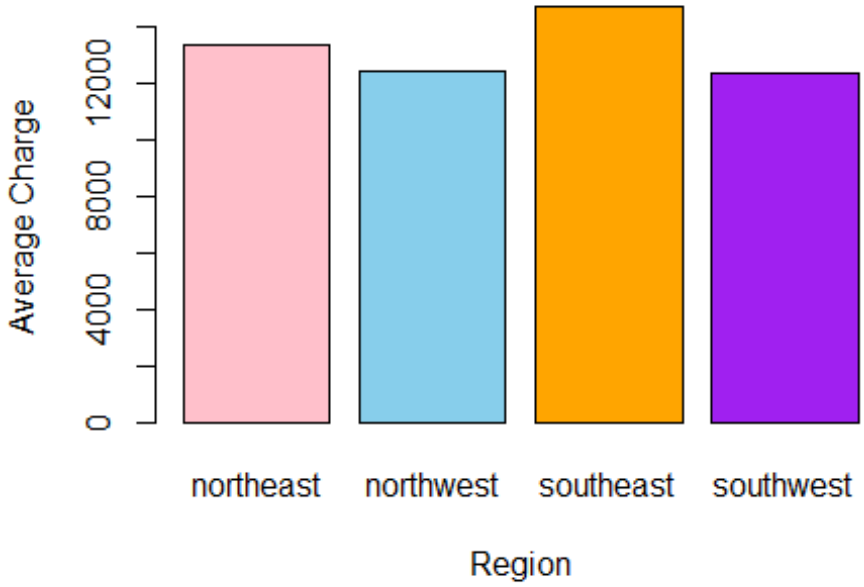
```
av_smoker_charge <- tapply(data$charges, data$smoker, mean)
barplot(av_smoker_charge, xlab = "Smoker", ylab = "Average Charge", col =
c("grey", "black"))
```



Comment: It is clearly evident from the plot that people who smoke tends to have a significantly higher average medical cost than people who don't.

Charges for poeple of different regions:

```
av_region_charge <- tapply(data$charges, data$region, mean)
barplot(av_region_charge, xlab = "Region", ylab = "Average Charge", col =
c("pink", "skyblue", "orange", "purple"))
```



Comment: Here also we see that there is not much difference in charges among people from different region.

Model Fitting

Before we fit any model into the data, at first we need to change the categorical columns into numerical ones so that we can work with them with ease.

```
data$sex <- as.numeric(factor(data$sex)) - 1
data$smoker <- as.numeric(factor(data$smoker)) - 1
data$region <- as.numeric(factor(data$region)) - 1
head(data)
```

```
##   age sex    bmi children smoker region   charges
## 1  19   0 27.900        0      1      3 16884.924
## 2  18   1 33.770        1      0      2  1725.552
## 3  28   1 33.000        3      0      2  4449.462
## 4  33   1 22.705        0      0      1 21984.471
## 5  32   1 28.880        0      0      1  3866.855
## 6  31   0 25.740        0      0      2  3756.622
```

We have now changed the whole dataset into numerical data and can now proceed with model fitting.

Split the data into training and testing

Here we allocate 80% of the data for training and the remaining 20% goes for testing.

```r
set.seed(82)
indices <- sample(nrow(data), 0.8*nrow(data))
train_data <- data[indices, ]
test_data <- data[-indices, ]
```

Linear Regression Model:

```r
lr_model <- lm(charges ~ ., data = data)
summary(lr_model)
```

```
##
## Call:
## lm(formula = charges ~ ., data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11343  -2811  -1017   1408  29751
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11811.45     955.72 -12.359  < 2e-16 ***
## age            257.20      11.90  21.616  < 2e-16 ***
## sex           -129.40     333.06  -0.389 0.697692
## bmi            332.60      27.73  11.993  < 2e-16 ***
## children       478.77     137.73   3.476 0.000525 ***
## smoker       23819.15     412.05  57.806  < 2e-16 ***
## region        -354.01     151.99  -2.329 0.020003 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6062 on 1330 degrees of freedom
## Multiple R-squared:  0.7506, Adjusted R-squared:  0.7494
## F-statistic:   667 on 6 and 1330 DF,  p-value: < 2.2e-16
```

Comment: From the summary we can see that the p-value for the coefficient of the variable 'sex' is almost 0.7 which is significantly high(>>0.05), specifying that it doesn't affect the response variable 'charges'. So now we will rebuild the model droping the variable 'sex'.

```r
lr_model2 <- lm(charges ~ . -sex, data = data)
summary(lr_model2)
```

```
##
## Call:
## lm(formula = charges ~ . - sex, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11403  -2811   -993   1405  29695
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11861.91     946.55 -12.532  < 2e-16 ***
## age            257.31      11.89  21.638  < 2e-16 ***
## bmi            332.08      27.69  11.992  < 2e-16 ***
## children       477.82     137.67   3.471 0.000535 ***
## smoker       23807.02     410.74  57.962  < 2e-16 ***
## region        -353.84     151.95  -2.329 0.020022 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6060 on 1331 degrees of freedom
## Multiple R-squared:  0.7505, Adjusted R-squared:  0.7496
## F-statistic: 800.9 on 5 and 1331 DF,  p-value: < 2.2e-16
```

Model Accuracy Check:

Now we will check the accuracy of the model and check how it performs with the test data

```r
lr_train_predicted <- predict(lr_model2, newdata = train_data)
lr_test_predicted <- predict(lr_model2, newdata = test_data)

rmse <- function(actual, predicted){
  sqrt(mean((actual - predicted)^2))
}

rsquared <- function(actual, predicted){
  sst <- sum((actual - mean(actual))^2)
  ssr <- sum((actual - predicted)^2)
  1 - (ssr/sst)
}

lr_rmse_train <- rmse(train_data$charges, lr_train_predicted)
lr_rsquared_train <- rsquared(train_data$charges, lr_train_predicted)

lr_rmse_test <- rmse(test_data$charges, lr_test_predicted)
lr_rsquared_test <- rsquared(test_data$charges, lr_test_predicted)

cat("Trainig RMSE: ", lr_rmse_train, "\nTraining R-squared: ",
```

```
lr_rsquared_train, "\n\nTesting RMSE: ", lr_rmse_test, "\nTesting R-squared:
", lr_rsquared_test, "\n")

## Trainig RMSE:  6149.95
## Training R-squared:  0.7438109
##
## Testing RMSE:  5614.047
## Testing R-squared:  0.776709
```

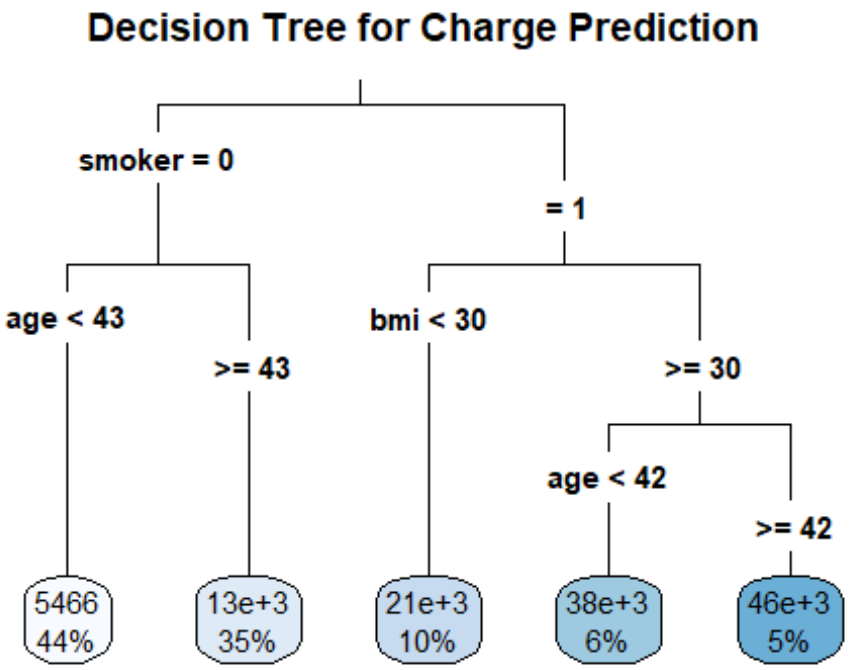
Decision Tree Regression Model:

First we load the library 'rpart' to perform Decision Tree regression

```
library(rpart)
library(rpart.plot)
```

Now we train our decision tree model with the train data and visualize it

```
dt_model <- rpart(charges ~ ., data = train_data, method = "anova")
rpart.plot(dt_model, type = 3, main = "Decision Tree for Charge Prediction")
```



Model Accuracy Check:

Now we check the performance of the model with the test data

```
dt_train_predicted <- predict(dt_model, newdata = train_data)
dt_test_predicted <- predict(dt_model, newdata = test_data)
```

```
dt_rmse_train <- rmse(train_data$charges, dt_train_predicted)
dt_rsquared_train <- rsquared(train_data$charges, dt_train_predicted)

dt_rmse_test <- rmse(test_data$charges, dt_test_predicted)
dt_rsquared_test <- rsquared(test_data$charges, dt_test_predicted)

cat("Training RMSE: ", dt_rmse_train, "\nTraining R-squared: ",
dt_rsquared_train, "\n\nTesting RMSE: ", dt_rmse_test, "\nTesting R-squared:
", dt_rsquared_test, "\n")
```

```
## Training RMSE:  4988.915
## Training R-squared:  0.8314108
##
## Testing RMSE:  4472.26
## Testing R-squared:  0.8582989
```

Random Forest Regression Model:

First we load the library 'randomForest' to perform random forest regression

```
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.
```

Now we train the random forest model with the train data

```
rf_model <- randomForest(charges ~ ., data = train_data)
print(rf_model)
```

```
##
## Call:
##  randomForest(formula = charges ~ ., data = train_data)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 23814785
##                     % Var explained: 83.87
```

Model Accuracy Check

```
rf_train_predicted <- predict(rf_model, newdata = train_data)
rf_test_predicted <- predict(rf_model, newdata = test_data)

rf_rmse_train <- rmse(train_data$charges, rf_train_predicted)
rf_rsquared_train <- rsquared(train_data$charges, rf_train_predicted)

rf_rmse_test <- rmse(test_data$charges, rf_test_predicted)
rf_rsquared_test <- rsquared(test_data$charges, rf_test_predicted)
```

```
cat("Training RMSE: ", rf_rmse_train, "\nTraining R-squared: ",
rf_rsquared_train, "\n\nTesting RMSE: ", rf_rmse_test, "\nTesting R-squared:
", rf_rsquared_test, "\n")
```

```
## Training RMSE:  3126.354
## Training R-squared:  0.9337945
##
## Testing RMSE:  4191.63
## Testing R-squared:  0.8755242
```

Conclusion

After fitting and checking the performance of all the models, We can see that the accuracy in training is 74%, 83%, and 93% for linear regression, decision tree and random forest model, specifying the superiority of the random forest model over the other two models while training with the train data. On the other hand in terms of predicting, the accuracy of the linear regression model and the decision tree model slightly increases to 77% and 86% respectively, while the accuracy of the random forest model decreases to 87% which is still better than the other two models.